

Spectral Clustering with Epidemic Diffusion

Laura M. Smith
USC Information Sciences
Institute

Kristina Lerman
USC Information Sciences
Institute

Cristina Garcia-Cardona
Claremont Graduate
University

Allon G. Percus
Claremont Graduate
University

Rumi Ghosh
HP Labs

ABSTRACT

Epidemic diffusion on a graph is a dynamic process that transitions simultaneously to all of a node's neighbors, in contrast to a random walk, which selects only a single neighbor. Epidemic diffusion is described by the replicator operator, an analog of the graph Laplacian that describes the behavior of random walks. We study the properties of the replicator operator. We show that the replicator is equivalent to the symmetric normalized Laplacian on a graph with edges reweighted by the eigenvector centralities of their incident nodes. Thus, more weight is given to edges connecting more central nodes. We propose a spectral clustering method, partitioning the nodes based on the componentwise ratio of the replicator's second eigenvector to the first. We compare the performance of our clustering technique to traditional spectral clustering methods on a variety of real world and synthetic graphs. We demonstrate how the replicator gives preference to cliques and clique-like structures, enabling it to more effectively discover communities that may be obscured by dense intercommunity linking.

1. INTRODUCTION

Clustering is one of the most widely used tools in data analysis. It is used to partition a set of items into groups of similar, or similarly behaving, items. When items are nodes on a graph, with edges linking similar or related items, clustering can be framed as a graph partitioning problem: namely, find groups of nodes with few edges between groups and many edges within groups. Graph partitioning has been used in a wide array of distinct applications, including image segmentation [27], data mining [2], and community detection in social networks [17].

Spectral clustering is one of the most intensely studied methods for graph partitioning [6, 22, 28, 30]. Spectral clustering relies on eigenvectors of the graph Laplacian matrix or its normalized versions. The eigenvectors associated with

the k smallest eigenvalues of the Laplacian can be used to partition the graph into k components [28, 30]. Spectral clustering performs well in many settings by solving a linear problem that is easy to implement, making it an attractive option for clustering data.

Spectral clustering is closely associated with random walks on graphs, a dynamic process that is the basis for diffusion. A random walk is a stochastic process that transitions from a node to a random neighbor of that node, and its dynamics are described by the (normalized) graph Laplacian. Spectral clustering partitions the graph in such a way that a random walk spends a long time within one cluster and seldom jumps to another cluster [27, 25]. Conversely, when a good graph partition exists, it will take a long time for a random walk to reach its equilibrium distribution [13].

Epidemic diffusion represents another type of a dynamic process that can take place on a graph. An epidemic is a process that transitions simultaneously to all the neighbors of a given node, rather than a single neighbor. It is often used to model the spread of a virus, or innovation, through a social network [1, 24]. Recently, Lerman and Ghosh introduced the replicator matrix [16], an analog of the graph Laplacian, that describes epidemic diffusion on graphs. They used the replicator to simulate dynamics of synchronization in a network of oscillators and showed that oscillators coupled via epidemic diffusion synchronize into different communities than diffusively coupled oscillators, whose dynamics are described by the graph Laplacian.

In this paper we analyze the properties of the replicator and use it for spectral clustering. Specifically, we show that the replicator is equivalent to the normalized symmetric Laplacian of a reweighted graph, where new edge weights are the product of old edge weights and the eigenvector centralities of the two end points. The eigenvector centrality [3] of a graph is given by the eigenvector corresponding to the largest eigenvalue of the adjacency matrix, or equivalently (by construction), the steady state of the replicator. This equivalence allows us to exploit some of the well-known relationships between spectral clustering and graph partitioning. To use the replicator for spectral clustering, we present and motivate a new approach that assigns nodes to clusters based on the componentwise ratio of the second to first eigenvectors. Given N nodes, this method leads to a computationally efficient procedure based on evaluating a quality

measure at $N - 1$ possible partitions of a vector.

We apply replicator-based spectral clustering to partition synthetic and real-world graphs with known community structure and compare the results to traditional spectral clustering that uses graph Laplacian matrices. We demonstrate that replicator-based clustering leads to a better recovery of ground truth on benchmark graphs, especially those that are more challenging for the Laplacian.

We make the following contributions in this paper: we

- analyze spectral properties of epidemic diffusion on graphs,
- demonstrate equivalence of the replicator and the symmetric normalized Laplacian of the reweighted graph,
- introduce a procedure for spectral clustering using the replicator matrix, and
- compare performance of spectral clustering using the replicator and the Laplacian on a variety of graphs.

Our work suggests that epidemic diffusion can be a useful probe of graph structure as it can illuminate properties of graphs that are distinct from those found by methods based on the random walk.

2. BACKGROUND AND RELATED WORK

An unweighted graph $G = (V, E)$, with vertices (or nodes) V and edges E , can be represented by a $|V| \times |V|$ matrix \mathbf{A} , called the adjacency matrix. Here, $A_{ij} = 1$ if $(i, j) \in E$, and $A_{ij} = 0$ otherwise. Also, we use the convention $A_{ii} = 0$. We consider undirected graphs, where $A_{ij} = A_{ji}$. The degree of node i is defined as the number of edges incident on it, $d_i = \sum_j A_{ij}$. Other useful constructs are \mathbf{D} , a diagonal degree matrix where $D_{ii} = d_i$, and the identity matrix \mathbf{I} .

2.1 Graph Laplacian and Spectral Clustering

The graph Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The eigenvalues and eigenvectors of \mathbf{L} capture many properties of the graph. In the simplest case, if the graph has k disjoint components, the first k smallest eigenvalues of \mathbf{L} are zero, and the corresponding leading eigenvectors are indicator functions assigning nodes to their respective cluster or community [30]. Even if the k smallest eigenvalues are not all zero, their corresponding eigenvectors can be used to partition nodes into k clusters by projecting these nodes onto a subspace of the first k eigenvectors and using standard clustering techniques such as k -means [22, 27]. The simplest spectral clustering method, spectral bisection, partitions nodes based on the values of the second eigenvector \mathbf{v} of the adjacency matrix or the graph Laplacian. A splitting value c is used to divide the nodes into different clusters based on whether $\mathbf{v}_i < c$ or $\mathbf{v}_i \geq c$ [28]. A range of splitting values have been used, including zero, the median value within the vector, the largest gap, and the value producing the best ratio cut, best conductance [14], or other measure.

In practice, normalized versions of the graph Laplacian produce better results in spectral clustering applications [22, 2]. Two examples are the symmetric normalized Laplacian $\mathbf{L}_s = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ and the random walk Laplacian

$\mathbf{L}_{rw} = \mathbf{I} - \mathbf{D}^{-1} \mathbf{A}$, so named because the transition probability of a random walk on a graph is given by $\mathbf{D}^{-1} \mathbf{A}$.

2.2 Graph Cuts and Their Quality Measures

Intuitively, a cluster is a set of nodes $S \subset V$ that are more tightly connected to other nodes within the cluster than to nodes outside of the cluster. We use $\bar{S} = V \setminus S$ to denote the complement of S , which consists of nodes that are not in S . In order to partition the graph into disjoint clusters, one typically wants to minimize the number of cut edges between partitions,

$$cut(S, \bar{S}) = \sum_{i \in S, j \in \bar{S}} A_{ij},$$

while maximizing cluster size, which may be measured by the number of nodes it contains, $|S|$, or the sum of the degrees of the nodes in the set, $vol(S) = \sum_{i \in S} d_i$.

Several functions have been proposed for measuring the quality of a graph cut. The best known of these are conductance $\phi(S)$, ratio cut $RCut(S)$, and normalized cut $NCut(S)$:

$$\phi(S) = \frac{cut(S, \bar{S})}{\min(vol(S), vol(\bar{S}))} \quad (1)$$

$$RCut(S) = \left(\frac{1}{|S|} + \frac{1}{|\bar{S}|} \right) cut(S, \bar{S}) \quad (2)$$

$$NCut(S) = \left(\frac{1}{vol(S)} + \frac{1}{vol(\bar{S})} \right) cut(S, \bar{S}). \quad (3)$$

There is a relationship between graph cuts and spectral clustering. Deciding which edges to cut to optimize any of these quality functions is an NP-complete problem. Spectral clustering solves a relaxation of the problem, where the discrete indicator variables that assign nodes to clusters become continuous. Although in general there are no useful bounds for the approximation produced by this relaxation [30], in practice it often provides a simple and effective clustering method. Solutions to the relaxed optimization problem are given by the second eigenvector of the graph Laplacian \mathbf{L} or the normalized graph Laplacian \mathbf{L}_s [28]. Relaxing $RCut$ leads to spectral clustering using \mathbf{L} , while relaxing $NCut$ leads to spectral clustering using \mathbf{L}_s [27, 30].

Conductance is a popular choice for evaluating the quality of a partition and has been used to identify communities in social and information networks [6]. Although conductance fails to identify communities in certain real-world networks, where they blend in by linking heavily to the rest of the network [17], it is nevertheless a valuable basis for comparison.

2.3 Spectral Clustering and Random Walks

There exists a further relationship between spectral clustering, the partition quality function, and properties of random walks. A random walk on a graph is a stochastic process that transitions to a randomly chosen neighbor of a given node. Cluster properties of the graph can be expressed in terms of the transition matrix of a random walk $\mathbf{D}^{-1} \mathbf{A}$ [18]. Spectral clustering finds a partition such that a random walk stays within the same cluster for a long time and seldom transitions between clusters [27, 25]. Therefore, the presence of a good partition (e.g., low conductance cut) implies that it

will take a random walk a long time to reach its equilibrium distribution.

3. EPIDEMIC DIFFUSION ON GRAPHS

An epidemic is a dynamic process with some probability of transitioning simultaneously to every neighbor of the current node. Such processes are used to model the spread of disease [12] and innovation [24] in social networks. Epidemics differ from random walks in important ways. First, rather than choosing a single neighbor to transition to or “infect” as the random walk does, an epidemic will attempt to “infect” every neighbor of a node. During a random walk, the probability to find the walker anywhere on the graph remains constant, and the random walk transition matrix is a stochastic matrix. Epidemics, alternatively, are locally non-conservative processes [16]. One can think of an epidemic as replicating itself with each successful transmission.

Lerman and Ghosh [16] introduced the replicator operator $\mathbf{R} = \lambda_{max} \mathbf{I} - \mathbf{A}$ to describe dynamics of nodes coupled via epidemic diffusion. λ_{max} is the largest eigenvalue of \mathbf{A} , also known as the epidemic threshold [31]. In this system, a dynamic variable u_i associated with node i can change its value based on the values of its neighbors according to:

$$\frac{du}{dt} = -\mathbf{R}u, \quad (4)$$

where \mathbf{R} replaces the Laplacian used in the analogous heat equation that gives the (diffusive) evolution of a random walk on a graph [5]. By construction, the replicator has a steady state given by $\boldsymbol{\theta}$, the eigenvector of \mathbf{A} associated with λ_{max} : $\mathbf{A}\boldsymbol{\theta} = \lambda_{max}\boldsymbol{\theta}$. $\boldsymbol{\theta}$ is also known as the *eigenvector centrality* [3], and it was introduced by Bonacich to explain the importance of actors in a social network based on the importance of the actors to which they were connected.

Clusters of nodes with similar values of the dynamic variable u emerge as the system of coupled nodes evolves towards the steady state [16]. This motivates a community detection method with nodes classified according to the rate of convergence to their steady-state values. For large time t , we approximate the solution to Eq. 4 using the two leading eigenvectors $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ of \mathbf{R} ,

$$\begin{aligned} u_i(t) &\approx c_1\theta_i + c_2e^{-\lambda_2 t}\psi_i \\ &= c_1\theta_i \left[1 + \frac{c_2}{c_1}e^{-\lambda_2 t}\frac{\psi_i}{\theta_i} \right], \end{aligned}$$

where c_1 and c_2 are constants, and λ_2 is the second smallest eigenvalue of \mathbf{R} associated with eigenvector $\boldsymbol{\psi}$. Therefore, convergence depends on ψ_i/θ_i , the componentwise ratio of the second to first eigenvectors. Note that eigenvectors of \mathbf{R} corresponding to \mathbf{R} 's two smallest eigenvalues are the same as the eigenvectors of \mathbf{A} corresponding to \mathbf{A} 's two largest eigenvalues.

3.1 Replicator as the Laplacian of a Reweighted Graph

In a social network, one might expect nodes of high “importance” to serve as attractors of other nodes, resulting in communities forming around nodes with large eigenvector centrality values θ_i . In this section we propose a modification of our graph, converting the unweighted network

into a weighted one where weights are given by the product of the eigenvector centralities of an edge's end points: $\tilde{A}_{ij} = A_{ij}\theta_i\theta_j$. Moreover, we show that the replicator on the unweighted graph given by \mathbf{A} is in fact exactly equivalent to the symmetric normalized Laplacian of the reweighted graph given by $\tilde{\mathbf{A}}$.

In the reweighted graph, the degree of node i is given by

$$\tilde{d}_i = \sum_j A_{ij}\theta_i\theta_j = \theta_i \sum_j A_{ij}\theta_j = \lambda_{max}\theta_i^2.$$

For convenience, define $\boldsymbol{\Theta}$ as the diagonal matrix whose elements are the components of eigenvector $\boldsymbol{\theta}$, i.e. Θ_{ii} . Then, from \tilde{A}_{ij} and \tilde{d}_i above,

$$\tilde{\mathbf{A}} = \boldsymbol{\Theta}\mathbf{A}\boldsymbol{\Theta} \quad \text{and} \quad \tilde{\mathbf{D}} = \lambda_{max}\boldsymbol{\Theta}^2. \quad (5)$$

We can now write the symmetric normalized Laplacian of the reweighted graph:

$$\begin{aligned} \tilde{\mathbf{L}}_s &= \mathbf{I} - \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2} \\ &= \mathbf{I} - \left(\frac{1}{\sqrt{\lambda_{max}}} \boldsymbol{\Theta}^{-1} \right) \boldsymbol{\Theta}\mathbf{A}\boldsymbol{\Theta} \left(\frac{1}{\sqrt{\lambda_{max}}} \boldsymbol{\Theta}^{-1} \right) \\ &= \mathbf{I} - \frac{1}{\lambda_{max}} \mathbf{A} \\ &= \frac{1}{\lambda_{max}} \mathbf{R}. \end{aligned}$$

Hence, $\mathbf{R} = \lambda_{max}\tilde{\mathbf{L}}_s$.

The equivalence between random walks and epidemics at first appears surprising, because these processes are fundamentally so different. In random walks, the probability to find the walker on any node of the graph is conserved, while epidemics are non-conservative [16]. The intuition for this equivalence is the following. A node's eigenvector centrality gives the number of paths connecting it to other nodes in the graph [10]. Since the epidemic replicates with each transition, the product of eigenvector centralities of two nodes captures how much new spreading “quantity” is produced when the epidemic transitions along the edge linking these nodes. By encoding the amount of non-conservation in edge reweighting, this scheme allows the epidemic to be reduced to a simple random walk on the graph.

3.2 Quality Measure for the Replicator

The equivalence proved above allows us to leverage the properties of the symmetric normalized Laplacian, along with its relationship to graph partitioning, for epidemic diffusion. Since the replicator is simply \mathbf{L}_s on a graph that is reweighted according to the scheme in Eq. 5, spectral clustering using the replicator corresponds to a relaxation of *NCut* on this reweighted graph. The appropriate measure for assessing graph cut quality with the replicator is therefore *NCut* on the reweighted graph.

4. SPECTRAL CLUSTERING VIA EPIDEMIC DIFFUSION

We propose a clustering method based on epidemic diffusion. Our clustering method seeks to minimize *NCut* on the reweighted graph. This often results in a partition different from that found by traditional spectral clustering meth-

ods that attempt to minimize conductance, $RCut$, or $NCut$ on the original graph.

4.1 Motivating Example

We use a simple example to highlight the differences between traditional graph partitioning and those based on epidemic diffusion. Consider the toy graph in Figure 1. We expect a good partition to group node 6 with other nodes in its clique, rather than separate it from its clique. However, the cut (B) that minimizes conductance and $NCut$ groups node 6 with nodes 1–5 and assigns nodes 7–11 to the other cluster. There are multiple cuts that minimize $RCut$, including one that forms a community consisting of nodes 3–5.

Now we reweight the edges of the graph according to eigenvector centrality, using the prescription $\tilde{\mathbf{A}} = \Theta \mathbf{A} \Theta$. Node 6 has the highest centrality. Furthermore, nodes that belong to the clique have higher eigenvector centrality values than other nodes, making the edges linking node 6 to the rest of the clique more “expensive” to cut. Consequently, nodes 6–11 are grouped together by the preferred cut (A) for minimizing both $RCut$ and $NCut$ on the reweighted graph. (Minimizing the conductance on the reweighted graph gives preference to cut (B), placing node 6 with nodes 1–5 and not with the clique.) The quality measure values are given in Table 1. Epidemic diffusion thus tends to preserve cliques and clique-like structures.

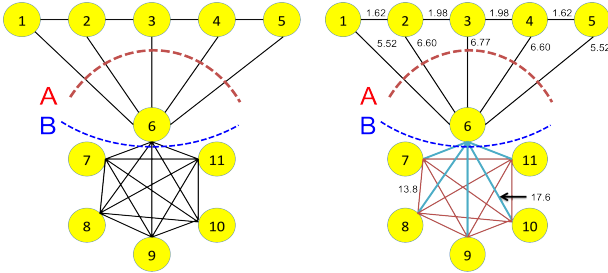


Figure 1: (Left) Toy graph. (Right) Reweighted toy graph with edge weights given by the product of eigenvector centralities of the edges’s two end-points. The red edges have weight 13.8, and the blue edges have weight 17.6. (Both) The possible optimal cuts are shown by the dotted curves A and B. (Best viewed in color)

Measure	Graph	Cut A	Cut B
Conductance ϕ	Original	0.385	0.217
$RCut$	Original	1.83	1.83
$NCut$	Original	0.528	0.417
Conductance ϕ	Reweighted	0.683	0.536
$RCut$	Reweighted	11.4	32.3
$NCut$	Reweighted	0.747	0.778

Table 1: Quality measures of cuts A and B for the graph and reweighted graph of Figure 1.

Tong et al. [29] propose a method for selecting which edges to delete to minimize the spread of an epidemic on a graph. Their method selects edges between nodes with highest eigenvector centralities. Our work provides an insight into the method. In the toy example, edges connecting node 6 to its

clique along cut (B) will be deleted by their method, thus preventing the epidemic from spreading to the clique. The larger the clique, the higher the eigenvector centrality score of its nodes; therefore, edges connecting the clique to the rest of the graph are more likely to be chosen for deletion.

4.2 Our Clustering Method

We propose a clustering method based on spectral bisection. First, we create a vector \mathbf{v} that is the componentwise ratio of the second eigenvector ψ to the first eigenvector θ of the operator \mathbf{R} and sort its values. Next, we examine all partitions given by the $N - 1$ possible cuts in this ordering and pick the partition that minimizes a quality measure [28]. The quality measures that we use with \mathbf{R} are conductance on the original graph and $NCut$ on the reweighted graph, which we denote “ $NCut$ (reweighted)” in the tables summarizing our results. We compare the resulting partitions with those produced by applying an analogous procedure to \mathbf{L} , with quality measures conductance and $RCut$, and \mathbf{L}_s , with quality measures conductance and $NCut$ (on the original graph). Additionally, for all three operators, we compare with the standard spectral clustering technique of k -means clustering on the first two eigenvectors corresponding to the smallest two eigenvalues of the operators [30], and the results are denoted in our tables as “ k -Means.”

Since our optimization procedure tests all $N - 1$ possible cuts within \mathbf{v} , it is far more exhaustive than k -means. It may seem that there would be some loss in accuracy from restricting our search to cuts in a one-dimensional projection, rather than searching over the entire subspace spanned by the first two eigenvectors θ and ψ . However, it has been observed [33, 27] that the componentwise ratio of the second to first eigenvector of \mathbf{L}_s is precisely equal to the second eigenvector of the random walk Laplacian \mathbf{L}_{rw} , whose first eigenvector is a constant vector. Thus, our algorithm is effective because it is a computationally efficient procedure for finding the best $NCut$ in the two-dimensional eigenspace of $\tilde{\mathbf{L}}_{rw}$, i.e., \mathbf{L}_{rw} on the reweighted graph. The advantages of using \mathbf{L}_{rw} in spectral clustering are discussed in [30].

5. EXPERIMENTAL RESULTS

In order to examine the differences in clustering using the operators \mathbf{L} , \mathbf{L}_s , and \mathbf{R} , we first return to the motivating example of Section 4.1. We then turn to communities where the ground truth is known. The real world examples that we will see are given in Section 5.2. To better understand the clustering mechanisms and the types of graphs each operator will perform the best, we construct random graphs with ground-truth communities. The description of these graphs and their results are provided in Section 5.3.

5.1 Motivating Example

Figure 1 displays the toy graph and the same network with reweighted edges. We apply our clustering methods using the operators \mathbf{L} , \mathbf{L}_s , and \mathbf{R} , splitting according to the operator’s corresponding quality measure, $RCut$, $NCut$, and $NCut$ on the reweighted graph. Also, we apply the clustering techniques to these same operators using the conductance quality measure on the original graph. All the resulting partitions are located in Table 2. The majority of the methods prefer cut (B), which splits the nodes 1–6 and 7–11.

Operator	Method	Class 1	Class 2
L	Conductance ϕ	1-6	7-11
L_s	Conductance ϕ	1-6	7-11
R	Conductance ϕ	1-6	7-11
L	$RCut$	Multiple	Ties
L_s	$NCut$	1-6	7-11
R	$NCut$ (reweighted)	1-5	6-11

Table 2: Partitions of graph in Figure 1 using our clustering methods. Note that only the bottom row uses a measure on the reweighted graph.

However, the replicator is the only one that places node 6 in the other community when minimizing with respect to the $NCut$ on the reweighted graph. This reflects the influence of reweighting the graph edges.

5.2 Real World Networks

A first test for using our clustering algorithm is to examine real world networks with identified communities. By applying our method to social networks, we are able to investigate the role of epidemic diffusion and the different operators in clustering. The data sets we present come from a network of monks in a monastery, a network of dolphins from New Zealand, and the U.S. House of Representatives.

To evaluate the resulting clustered communities, we use the Normalized Mutual Information (NMI) measure [7] and the Purity measure [20], both of which examine how similar the resulting partition is when compared to the ground truth communities. For both measures, a value of 1.0 is optimal.

5.2.1 Sampson’s Monastery

In 1969, Sampson observed a community of 18 monks at a monastery for one year [26]. During this time, a breakup of the monastery occurred when monks 2, 3, 17, and 18 were expelled from the monastery and monks 1, 7, 14, 15, and 16 voluntarily left immediately. Eventually, the only remaining monks were 5, 6, 9, and 11. At three times prior to this event, Sampson had the monks rank the top three individuals they liked. After the breakup, he recorded additional rankings to include dislike, esteem and disesteem, positive and negative influence, and praise and blame.

Using the ranking data, we create a social network of the monks and attempt to find the partition between those that left immediately, either voluntarily or by expulsion, and those that initially stayed. To do this, we first consider all edges that connect monks that mutually listed each other as one of their top 3 individuals during any of the liking rankings prior to the splitting. We then remove edges where either monk ranked its neighbor in the top 3 of the dislike, disesteem, negative influence, or blame categories. This will account for changes in attitude over the year. The resulting network is shown in Figure 2. For alternative methods of network construction, see e.g. [4].

From this network, we use our clustering methods with the goal of recreating the partition of individuals that stayed and those that left immediately. These results are provided in Table 3. Note that R has substantially higher metric values for this network, mislabeling only monks 12 and 13,

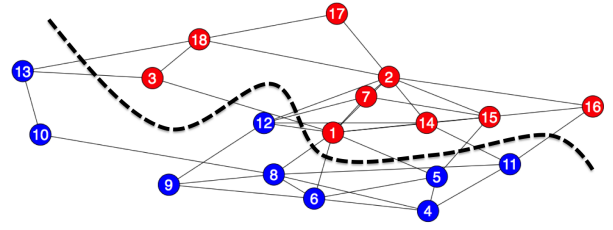


Figure 2: Sampson’s Monastery Network. The two groups are separated by colors and the dotted curve. The upper group gives the individuals that left immediately. (Best viewed in color.)

two monks that eventually left the monastery after some time had passed. The other operators perform poorly.

	Method	NMI	Purity
L	Conductance ϕ	0.0606	0.6111
L_s	Conductance ϕ	0.0606	0.6111
R	Conductance ϕ	0.0606	0.6111
L	$RCut$	0.0606	0.6111
L_s	$NCut$	0.0606	0.6111
R	$NCut$ (reweighted)	0.5926	0.8889
L	k -Means	0.0606	0.6111
L_s	k -Means	0.0606	0.6111
R	k -Means	0.5926	0.8889

Table 3: Results for partitioning the group of monks in a monastery using our clustering methods and k -means clustering on the first two eigenvectors.

5.2.2 Dolphin Network

A study by Lusseau et al. [19] examined the social behaviors of 62 dolphins in New Zealand. A network was created by linking two dolphins that were observed together and is shown in Figure 3. At some point during the study, the dolphins split into two groups when the individual, represented by the node with a white interior, departed.

While our method is not as successful in recovering the two communities, this graph is instructive for another reason. The node with the black interior is in the red community but is not connected to any red node other than the one that departed. This suggests that the network may be formed from incomplete observations, and indeed most partitioning methods misclassify this node. For example, the cuts for L and L_s , as well as R with conductance, have better performance (Table 4), but they all misclassify this node, as does the modularity method of [21]. When optimizing the reweighted $NCut$ with R , however, the reweighting forces it to remain in the proper group in spite of the possible sampling bias that may account for the replicator’s lower NMI and purity scores.

5.2.3 House of Representatives

The 98th United States House of Representatives voting data set from 1983–1985 provides information on how congressmen voted on 908 different issues [23]. This congress was comprised of 272 Democrats and 163 Republicans. We

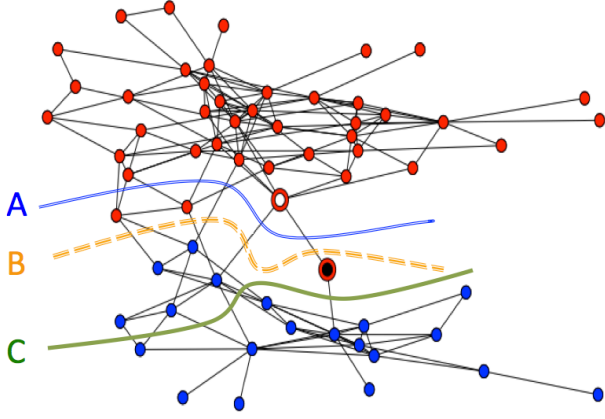


Figure 3: Dolphin Network with two groups indicated by color. The cuts represent the partitions selected by (A) the modularity method of [21], (B) the cuts for L and L_s , as well as R with the conductance quality measure, and (C) R with the reweighted $NCut$ quality measure. (Best viewed in color.)

	Method	NMI	Purity
L	Conductance ϕ	0.8888	0.9839
L_s	Conductance ϕ	0.8888	0.9839
R	Conductance ϕ	0.8888	0.9839
L	$RCut$	0.8888	0.9839
L_s	$NCut$	0.8888	0.9839
R	$NCut$ (reweighted)	0.6292	0.9194
L	k -Means	0.8888	0.9839
L_s	k -Means	0.8888	0.9839
R	k -Means	0.7769	0.9677

Table 4: Results for partitioning the dolphins using our clustering methods and k -means clustering on the first two eigenvectors.

construct a network for the 435 individuals by considering co-votes. If two individuals both vote the same on a measure, then we add 1 to the edge weight between the nodes. If they vote opposite of one another, then we add -1 to the edge weight. The final network is taken by thresholding the final edge weight between nodes, considering edges to exist only if the weight is greater than T . Politicians of separate parties may agree on some issues. Therefore, setting $T = 0$ would create a nearly complete graph. Instead, we set T equal to the average value of the final edge weights.

A subset of this dataset consisting of 16 votes has been used for clustering congressmen [2, 11]. The votes were chosen by the Congressional Quarterly Almanac as the key votes for this congress [9]. We create a network from these 16 votes with 47,772 edges. When applying our clustering methods to this network, we obtain the best results with L_s and R , minimizing with respect to the corresponding quality measures of $NCut$ on the original and reweighted graph, respectively. The purity scores are 0.8782 for L_s and 0.8759 for R .

However, selecting the key votes to use is a subjective process that eliminates much of the data. Additionally, an important subset of votes is generally not available. If instead

we use all 908 votes, we obtain the network shown in Figure 4. This network has a density of 0.5260 and average clustering coefficient of 0.8682. There are 51,033 edges in this network with 8,111 between the two parties.

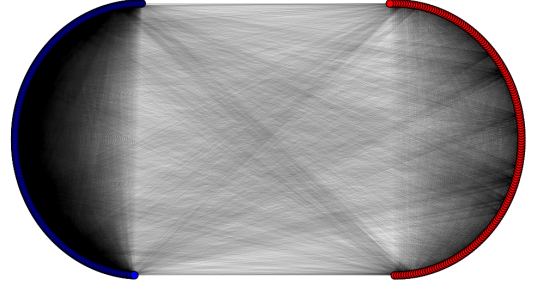


Figure 4: 98th Congress Network with 908 votes. Nodes on the left indicate democrats and nodes on the right indicate republicans. An edge is present if two individuals have an edge weight greater than the average weight of $T = 252.4$. (Best viewed in color.)

We use both the conductance metric and the respective quality measures for clustering this network with the different operators, attempting to divide the nodes into the two political parties. Table 5 shows the performance of the methods, indicating that the L operator works best with conductance, whereas the other two prefer their respective quality measures. Additionally, L_s and R give the same splitting when using the conductance quality measure. However, R performs the best overall with the reweighted $NCut$. Furthermore, the purity scores for L_s and R are significantly higher for the graph that used all 908 votes than the 16 votes.

	Method	NMI	Purity
L	Conductance ϕ	0.5346	0.8934
L_s	Conductance ϕ	0.5794	0.9184
R	Conductance ϕ	0.5794	0.9184
L	$RCut$	0.0032	0.6168
L_s	$NCut$	0.5988	0.9229
R	$NCut$ (reweighted)	0.6318	0.9297
L	k -Means	0.0002	0.6168
L_s	k -Means	0.5874	0.9206
R	k -Means	0.5327	0.9048

Table 5: Results for partitioning the 98th U.S. House of Representatives (from 908 votes) using our clustering methods and k -means clustering on the first two eigenvectors.

5.3 Synthetic Benchmarks

We use synthetic graphs to gain better insight into the characteristics of graphs for which different operators find better solutions. Lancichinetti and Fortunato proposed an algorithm to generate random graphs with known hierarchical structure [15]. The N nodes are divided into macro communities, which are themselves composed of micro communities, and then edges between nodes are created using mixing parameters μ_1 and μ_2 . The parameter μ_1 designates

the fraction of a node's edges that will connect to nodes in a different macro community, and μ_2 gives the fraction of edges that will connect to nodes in a different micro community within the same macro community. The remaining $(1 - \mu_1 - \mu_2)$ fraction of edges link to other nodes within the same micro and macro communities. These benchmark networks allow us to systematically explore the performance of different spectral clustering approaches.

Using software available on [8], we generated 100 networks for each set of parameter values. We took $N = 100$ with two macro communities. We varied μ_1 and μ_2 between 0 and 0.5. Our goal is to correctly identify the macro communities.

To better understand the properties of the generated random graphs, we look at the average clustering coefficient. The clustering coefficient for a node is given by taking the number of its neighbors that are connected by an edge divided by the total number of possible triangles. The average clustering coefficient takes the mean value for all nodes. A complete graph has an average clustering coefficient of 1, and a graph with no edges between nodes has a value 0. The heatmap in Figure 5 shows the mean average clustering coefficient over the 100 runs for fixed parameters, with μ_1 and μ_2 ranging between 0 and 0.5. The mean clustering coefficient ranges between 0.23 and 0.6421. This indicates the synthetic graphs have graph properties similar to those often found in real world social networks [32].

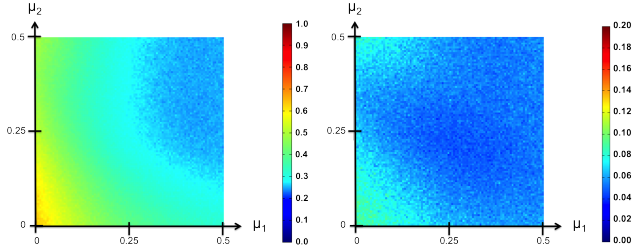


Figure 5: Each pixel represents the mean average clustering coefficient (left) and the standard deviation (right) across 100 runs for fixed (μ_1, μ_2) . (Best viewed in color.)

5.3.1 Results with the Conductance Quality Measure

Figure 6 shows the results for minimizing the conductance quality measure in splitting. Each pixel in the left column of images represents the average NMI value across 100 runs for fixed parameter values (μ_1, μ_2) . Note that as μ_1 increases, the number of edges between the two communities increases, making it more difficult to determine the partitioning. The right column provides the standard deviation of NMI across the runs.

The main difference between the plots is that \mathbf{L}_s produces a higher average NMI score than \mathbf{L} as μ_1 increases. Additionally, \mathbf{R} produces a higher average NMI score than \mathbf{L}_s as μ_1 increases. Therefore, \mathbf{R} provides the best results of the three operators for a wider range of μ_1 . We also see that \mathbf{R} has substantially lower standard deviation than the other two operators, suggesting that the NMI scores are consistently higher. To highlight the regions where each operator performs the best, we take the operator(s) with the highest

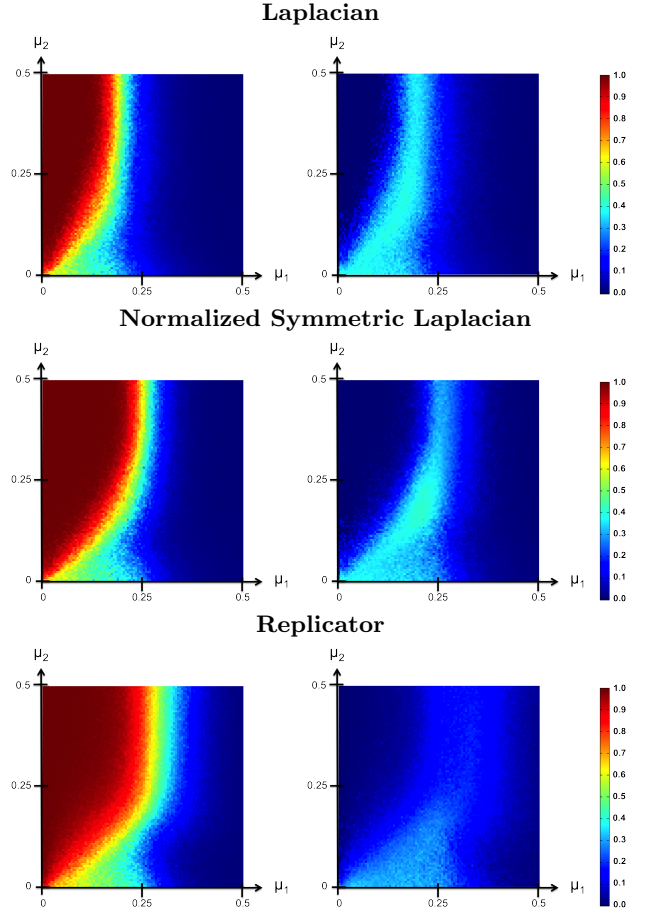


Figure 6: NMI scores for minimizing the conductance quality measure. Each pixel represents the average (left) or standard deviation (right) NMI score across 100 runs for fixed (μ_1, μ_2) . (Best viewed in color.)

average score at each point and give it a designated color. The color scale and best average value images are shown in Figure 7, and multiple listed operators indicate a tie. We observe that \mathbf{R} has the highest average NMI score for larger μ_1 values, and multiple operators tie for smaller μ_1 values. Since parameter μ_1 controls the fraction of edges connecting different communities, the replicator is better able to reconstruct the original communities in graphs where the community structure is obscured by intercommunity links.

5.3.2 Results with the Related Quality Measure

As seen in Section 4, the operators \mathbf{L} , \mathbf{L}_s , and \mathbf{R} are related to the minimization problems of $RCut$, $NCut$, and $NCut$ on a reweighted graph, respectively. In this section, we compare the results for each operator when minimizing their respective quality measures. We then calculate the average and standard deviation of the NMI scores for a fixed set of parameters and display the results in Figure 8. Included in Figure 7 is the image which identifies the operator(s) with the best average NMI score. For a tie, multiple operators are listed. We see similar results as when the partitioning was done with respect to the conductance quality measure.

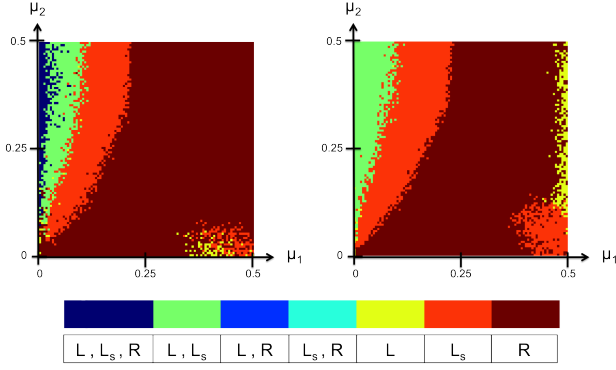


Figure 7: Best NMI with the conductance quality measure (Top Left) and the operators’ respective quality measures (Top Right). (Top) Each pixel represents the operator(s) with the best average NMI value across 100 runs for fixed parameter values (μ_1, μ_2) . (Bottom) Color scale for operators. Multiple operators indicate a tie in the average NMI value. (Best viewed in color.)

5.4 Analysis

The clustering methods using the replicator proposed in this paper perform well and often do better than traditional spectral clustering methods. The first example of Section 5.1 demonstrates how the replicator operator tends to favor cliques when clustering. For Sampson’s monastery graph, the replicator operator performed exceptionally well compared to the rest, only mislabeling two of the monks that most algorithms would have as well. The dolphin network results demonstrate the differences when minimizing with respect to the weighted $NCut$ and conductance, and all three operators give the same results with cut (B) when using the conductance quality measure, outperforming the modularity method. With a larger network, the U.S. House of Representatives’ voting data shows R has slightly better performance than L_s . Thus, when compared to known ground truth communities from the real world, our proposed clustering techniques are able to identify groups of individuals with like characteristics.

The synthetic benchmarks provide us with a general idea of the parameter regimes where each operator has its strengths. As the proportion of a node’s edges that connect to individuals in the opposite community, μ_1 , increases, it becomes more difficult to divide the network into the correct communities. Through the examples of Section 5.3, we find that L and L_s give better results when μ_1 is small (very few links between the two communities). As μ_1 increases, R dominates with a higher NMI score. Additionally, R has the lowest standard deviation of the three operators, indicating a consistent performance in identifying clusters.

6. CONCLUSION

Spectral clustering is traditionally done with the Laplacian, but this is one of many operators used. In this paper, we introduced a method for spectral clustering using the replicator, an operator describing epidemic diffusion on graphs. We have shown that this operator is equivalent to the normalized

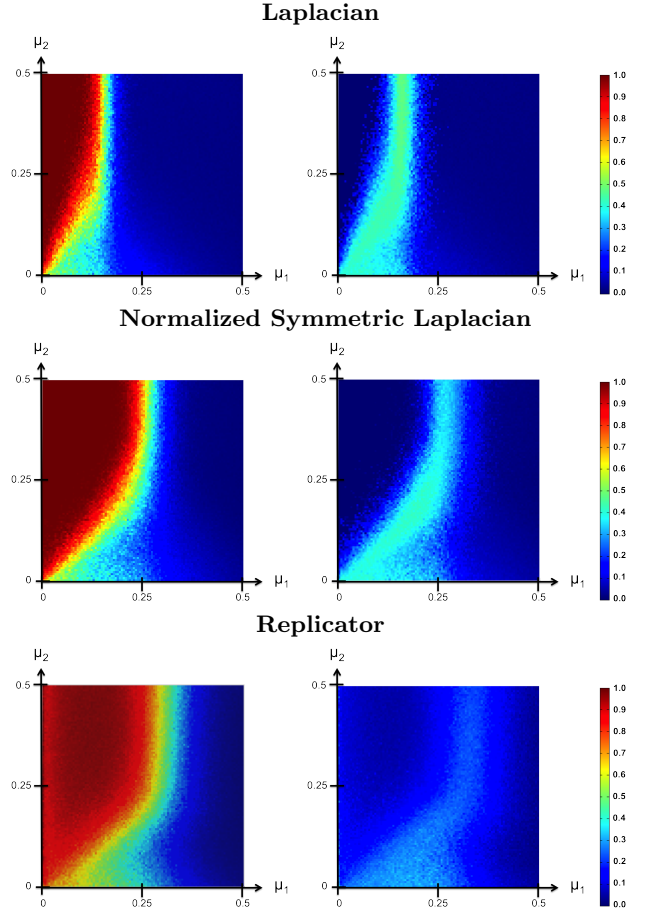


Figure 8: NMI scores for minimizing the respective operators’ quality measure. Each pixel represents the average (left) or standard deviation (right) NMI score across 100 runs for fixed (μ_1, μ_2) . (Best viewed in color.)

symmetric Laplacian on a graph with edges reweighted according to the eigenvector centrality measure. By reweighting the edges, a higher weight is placed on globally important nodes. Thus, this method tends to preserve cliques.

We introduced a spectral bisection approach based on the componentwise ratio of the second eigenvector to the first of R and chose the partition by splitting the sorted vector where a quality measure is minimized. We compared our method to spectral clustering based on L and L_s and showed that in many real-world and synthetic graphs with known community structure, the replicator was better able to recover the underlying community structure. In synthetic graphs, when more edges connect the two macro communities, the Laplacian and symmetric normalized Laplacian have a more difficult time identifying the communities. In these cases, the replicator gives the best performance. By reweighting the edges using eigenvector centrality, more importance is given to more central nodes. Thus, the edges that pass between regions are given less influence if they are not linking nodes of high centrality. This gives a better partition by limiting the cuts to influential edges.

Acknowledgements

The authors are grateful to Arjuna Flenner, Yves van Genip, and Blake Hunter for many instructive conversations and suggestions. KL and RG are also greatly indebted to Shanghua Teng, whose insights and enthusiasm continue to inspire them. This paper is based on work funded by the Air Force Office of Scientific Research under contracts FA9550-10-1-0569 and FA9550-10-1-0102, and by the National Science Foundation under grant 0915678.

7. REFERENCES

- [1] R. M. Anderson and R. May. *Infectious diseases of humans: dynamics and control*. Oxford University Press, 1991.
- [2] A. Bertozzi and A. Flenner. Diffuse interface models on graphs for classification of high dimensional data. *Multiscale Modeling & Simulation*, 10(3):1090–1118, 2012.
- [3] P. Bonacich and P. Lloyd. Eigenvector-like measures of centrality for asymmetric relations. *Social Networks*, 23(3):191–201, 2001.
- [4] P. Bonacich and P. Lloyd. Calculating status with negative relations. *Social Networks*, 26(4):331–338, 2004.
- [5] F. Chung. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences*, 104(50):19735–19740, Dec. 2007.
- [6] F. R. K. Chung. *Spectral Graph Theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. American Mathematical Society, Feb. 1996.
- [7] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005.
- [8] S. Fortunato. Benchmark graphs to test community detection algorithms, October 2011. <https://sites.google.com/site/santofortunato/inthepress2>.
- [9] A. Frank and A. Asuncion. UCI machine learning repository, 2010.
- [10] R. Ghosh and K. Lerman. Parameterized centrality metric for network analysis. *Physical Review E*, 83(6):066118, June 2011.
- [11] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. *Data Engineering, International Conference on*, 0:341–352, 2005.
- [12] H. W. Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42(4):599–653, 2000.
- [13] M. Jerrum and A. Sinclair. Conductance and the rapid mixing property for markov chains: the approximation of permanent resolved. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, STOC ’88, pages 235–244, New York, NY, USA, 1988. ACM.
- [14] R. Kannan and S. Vempala. *Spectral Algorithms*, volume 4 of *Foundations and Trends in Theoretical Computer Science*. Publishers Inc., Hanover, MA, 2009.
- [15] A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E*, 80(1), 2009.
- [16] K. Lerman and R. Ghosh. Network structure, topology and dynamics in generalized models of synchronization. *Physical Review E*, 86(026108), 2012.
- [17] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. 2008.
- [18] L. Lovász. *Random Walks on Graphs: A Survey*, pages 353–397. 1993.
- [19] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associates. can geographic isolation explain this unique trait? *Behavioral Ecology and Sociobiology*, 54:396–405, 2003.
- [20] C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.
- [21] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3), 2006.
- [22] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, volume 14, pages 849–856, 2001.
- [23] K. Poole. Voteview website. <http://voteview.com/house98.htm>.
- [24] E. M. Rogers. *Diffusion of Innovations*, 5th Edition. Free Press, 5th edition, Aug. 2003.
- [25] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, Jan. 2008.
- [26] S. Sampson. *Crisis in a cloister*. PhD thesis, Cornell University, 1969.
- [27] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [28] D. A. Spielman and S.-H. Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications*, 421(2-3):284–305, Mar. 2007.
- [29] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos. Gelling, and melting, large graphs by edge manipulation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM ’12, pages 245–254, New York, NY, USA, 2012. ACM.
- [30] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, Dec. 2007.
- [31] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos. Epidemic spreading in real networks: an eigenvalue viewpoint. In *Proceedings of the 22nd Symposium on Reliable Distributed Systems*, pages 25–34, Los Alamitos, CA, USA, Oct. 2003. IEEE.
- [32] D. J. Watts. Networks, dynamics, and the small-world phenomenon. *The American Journal of Sociology*, 105(2):493–527, 1999.
- [33] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *Proceedings IEEE International Conference on Computer Vision*, pages 975–982, 1999.